SUN ZFS STORAGE APPLIANCE

Oracleテクニカル・ホワイト・ペーパー
2012年11月

ストレージのパフォーマンスを最大化するパーティション・アライメント

目次

概要 4
ハード・ディスクの使用準備 6
ディスクが動作する仕組み 6
ディスクのアドレッシング方法 7
ハード・ディスクのインタフェース 7
Advanced Technology Attachment（ATA） 8
Serial ATA（SATA） 8
Small Computer System Interface（SCSI） 8
Serial Attached SCSI（SAS） 8
ファイバ・チャネル（FC） 8
iSCSI 8
ストレージのナチュラル・ブロック・サイズ 9
ディスク・ドライブへのパーティションの適用 10
パーティション化の標準の変化 10
標準の変化がパーティション・ツールとアライメントに及ぼす影響 11
1つのディスク/LUNでの複数のパーティションの使用 13
ディスクの準備とパーティション化に関する考慮事項 14
直接接続型ディスクと比較したストレージ・サブシステムのLUNのアライメント 15
Sun ZFS Storage ApplianceでのLUNの使用準備 16
FC環境でのSun ZFS Storage Appliance 16
iSCSI環境でのSun ZFS Storage Appliance 18
Oracle Solarisでのディスク管理 20
x86ベースのシステムでのOracle Solaris 22
SMIディスク・ラベルの使用 22
EFIディスク・ラベルの使用 24
SPARCでのOracle Solaris 26

SMIディスク・ラベルの使用 26
EFIディスク・ラベルの使用 27
Oracle Solarisのパーティション・アライメントの概要 28
Oracle Solarisのクライアントに関する推奨事項 28
Oracle SolarisホストOSにおけるSun ZFS Storage ApplianceのLUN構成のヒント 28
Linuxでのディスク管理 30
推奨事項 31
Oracle VMを使用したディスク管理 33
Oracle Automatic Storage Managementを使用したディスク管理 34
Oracle Automatic Storage Managementを使用したアライメントに関する考慮事項 35
Microsoft Windowsオペレーティング・システムを使用したディスク管理 36
Microsoft Windows Server 2008およびWindows Vista 36
Microsoft Windows Server 2003以前 36
VMwareを使用したディスク管理 37
VMware VMFSのブロック・サイズ 38
VMwareとNFS 38
Sun ZFS Storage Applianceの分析ツールを使用したミスアライメントの診断 40
結論 42
付録：参考資料 43

図目次

図1. ハード・ディスク・ドライブのコンポーネント.....6
図2. ミスアライメントされたパーティションの例 12
図3. Microsoft Windowsベースのオペレーティング・システムでの fdiskパーティションのアライメントの例.....13
図4. Sun ZFS Storage ApplianceでのLUNの作成.....16
図5. Sun ZFS Storage Applianceの冗長SAN構成.....17
図6. Sun ZFS Storage ApplianceのBUIに表示されたLAN IPMPグループ ...18
図7. Sun ZFS Storage ApplianceのBUIに表示されたLAN IPMP グループの設定.....18
図8. x86システムでOracle Solarisを使用する場合の、fdiskと SMI VTOCのデフォルト・レイアウト 22
図9. Oracle SolarisでのデフォルトのEFIパーティション・レイアウト 24
図10. VMwareデフォルトのVMFS3パーティション・レイアウト.....37
図11. Sun ZFS Storage ApplianceのBUIを使った共有レコード・ サイズの設定 39
図12. それぞれセクター256と34で始まる、EFIラベル付きLUNの パーティションのI/O表示.....41

概要

このホワイト・ペーパーでは、論理LUNまたはハード・ディスクを正しくパーティション化することの重要性について説明します。

アプリケーション・パフォーマンスを最適化するようにディスクとストレージ・サブシステムを構成するのは、多くの場合、芸術であると同時に技術であり、エキスパートに委ねるのが一番であるとみなされています。ただし、パーティションのアライメントはある程度の基礎知識があれば十分であり、構成を正しく設定すれば、I/Oパフォーマンスが大幅に低下するリスクを回避できます。ストレージ・サブシステムとベスト・プラクティスのドキュメントでは、ディスクのパーティションを基盤となるストレージ・ハードウェアのブロック構造とアライメントするというごく基本的な手順が見落とされていることが多いです。

コンピュータ・システムでストレージ・ユニットを使用できるようにするには、ストレージ・ユニットをソフトウェア・パーティション・ツールで初期化する必要があります。このソフトウェア・パーティション・ツールは、コンピュータ・システムにインストールされている特定のオペレーティング・システム・ソフトウェアに含まれています。

パーティショニング・ソフトウェアによって、オペレーティング・システムで使用される論理ストレージ・ユニットが作成されます。ほとんどのパーティショニング・ソフトウェアは、まだなお、トラックが含まれている物理プラッタの512バイト（B）のサイズのセクターをディスクが使用するという基準に基づいて動作します。パーティションは、シリンダ、ヘッド、セクターのアドレッシング・メカニズムを使用して、シリンダ単位で表される開始境界と終了境界で作成されます。ただし、新世代のディスク・ドライブとソリッド・ステート・ドライブ（SSD）では、サイズの大きいセクターが使用され、シリンダあたりのセクター数は固定ではありません。ファイバ・チャネル・プロトコルやiSCSIプロトコルで表される論理ユニットでは、セクターとシリンダの概念は適用されなくなっており、論理ブロック・アドレス（LBA）は元のセクター・サイズ512Bを大きく上回るブロック・サイズで使用されています。

パーティション・ツールでまだなお512Bのセクターベースのアドレッシング・スキームを使用している場合、パーティションの開始位置は基盤となるディスクまたはLUNの論理ブロックの開始位置には配置されない可能性が大きいです。このことが、ディスクまたはLUNとのI/Oパフォーマンス全体に大きく影響する可能性があります。1つのブロックに対する論理書込みが、2つのディスク・ブロックに対する物理読取り/変更/書込み操作となる可能性があります。そのため、パーティションの作成時に、ディスクまたはLUNに作成される各パーティションの開始位置が、ディスクの物理セクターまたはLUNの論理ブロックの開始位置とアライメントされていることを確認することが重要です。パーティションを作成したら、開始位置を変更するには、作成したパーティションを削除する必要があります。この場合、そのパーティションに格納されているデータが失われます。

このホワイト・ペーパーでは、パーティションの構造、およびディスク・ドライブのセクターまたはLUNのLBAにパーティションを確実にアライメントする方法について詳しく説明します。また、Sun ZFS Storage Applianceで提供されるLUN（およびLUNのLBAサイズ）を最適に構成する方法についても説明します。

最後に、Sun ZFS Storage Applianceで提供されるLUNを最適に利用するために、さまざまなベンダーのオペレーティング・システムでパーティション・ツールを使用する方法について説明します。

ハード・ディスクの使用準備

オペレーティング・システムでハード・ドライブを使用する前に、いくつかの基本情報でドライブを初期化する必要があります。この情報は、パーティション情報、オプションのブート・ソフトウェアの場所、およびブート・ソフトウェア自体で構成されます。この情報に使用されるサイズと場所は、セクター、トラック、シリンダまたはLBAエンティティの形式でディスク・ドライブのハードウェア特性で表されます。

これらの特性について説明するため、以降のセクションでは、ディスク・ドライブの物理的な側面、およびディスク・ドライブ内のデータを編成する方法について説明します。

ディスクが動作する仕組み

コンピュータのディスク・ストレージ・システム（305 RAMAC）は1956年にIBMによって導入されました。この305 RAMACは、直径24インチの5MBのディスクを備えていました。マイクロコンピュータに対応した初のディスク・ドライブが導入されて、ディスクがさらに広く使用されるようになったのは、1980年でした。

この間、使用されているテクノロジーはそれほど変化していません。ディスク・ドライブは、1つまたは複数のプラッタ、ヘッド、および制御電子回路の3つの主要コンポーネントで構成されています。

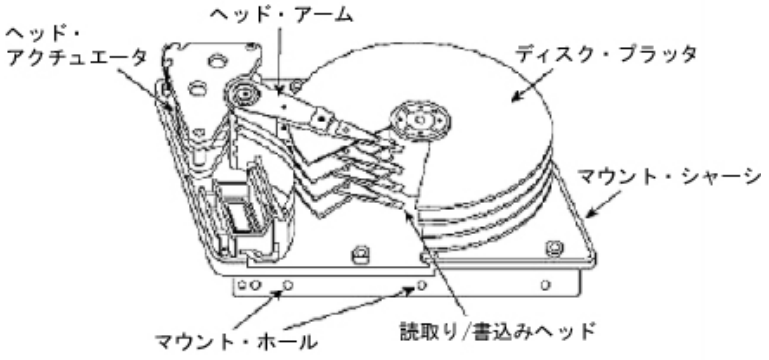

図1. ハード・ディスク・ドライブのコンポーネント

複数のプラッタが、1つのスピンドルによって積み重ねられて保持されています。読取りヘッドと書込みヘッドは各プラッタの上に配置されています。プラッタの表面には、データを格納するのに使用する磁性材料がコーティングされています。各プラッタは、データを格納する非常に細い同心円である多数のトラックに分割されています。各トラックは多数のセクターに分割されています。これらのセクターのサイズは常に512Bでしたが、現在では特にソリッド・ステート・ドライブで4KBのサイズのセクターも使用されています。

ディスクのアドレッシング方法

各プラッタの同じ位置にあるすべてのトラックをシリンダといいます。データは、シリンダ番号、ヘッド番号、およびセクター番号で配置されます。ディスク・ドライブの制御電子回路がヘッドを適切なシリンダに配置し、適切なヘッドをアクティブ化し、セクターとの間でデータの読取りと書込みを実行できるように、受け渡す適切なセクターを待機します。この場所のアドレッシング形式をCHS（シリンダ、ヘッド、セクター）またはジオメトリベースのアクセスといいます。

このデータ・ストレージの編成構造の他に、ディスク・ドライブは、データ・トラックにヘッドを適切に配置するためにドライブ・メーカーによって配置された構造も備えています。この構造の詳細については、ここでは説明していません。

初期設計で、各トラックには同じ数のセクターが配置されます。情報の密度はディスク内側のトラックがもっとも高く、これにより、使用するセクターの最大数が決まります。

ディスクの容量を増やすようにデータ転送速度が可変の記録手法を導入すると、外側のトラックで内側のトラックよりも多くのセクターを使用できるようになります。ディスクの電子回路では、変換スキーマを使用して、トラックあたり一定数のセクターがディスクに配置されているように見えるようにしていました。この方法で、ジオメトリベースのディスク・アクセス方法が引き続き機能していました。

多くのコンピュータ・アーキテクチャでは、低レベルのI/Oインタフェース・ソフトウェア・レイヤーを使用して、ハードウェア固有のインタフェースをコンピュータOSから抽象化しています。BIOS（基本入出力システム）と呼ばれるこのソフトウェアは、コンピュータのハードウェアのチップに配置されています。このBIOSによってディスク・ドライブのジオメトリが決まり、ジオメトリタイプのディスク・アクセスに基づいてドライブを操作する一連の機能がOSに提供されます。

ディスク容量が増加し、ディスク内部のデータ・アドレッシング構造がさらに複雑になると考えると、現在のコンピュータのBIOSでは固定のCHSマッピング・スキーマを使用できなくなります。ディスク・サイズはブート時に決まり、論理ブロック・アドレス（LBA）スキーマが使用されます。シリンダ数、ヘッド数、およびセクター数を参照するのではなく、ディスクは、サイズが512Bの各ブロックで構成されて連続したブロック番号が付いた1つのユニットとして認識されます。現在のディスク・ドライブはすべて、SCSI仕様の既知のアドレッシング形式であるLBAをサポートしています。必要に応じて、BIOSのI/O機能によって、LBAアドレッシングとCHSアドレッシングの間で変換が実行されます。

最新のオペレーティング・システムでは、ディスクへのアクセスにLBA直接アクセス方法を使用する、独自のディスクI/Oレイヤーが使用されています。

ハード・ディスクのインタフェース

ハード・ディスク・ドライブは、ディスクをコンピュータ・システムに接続するために、さまざまなインタフェースを備えており、さまざまなタイプのバス接続テクノロジーが使用されています。このテクノロジーは、データと制御が別々の接続から1つのシリアル・タイプの接続へと進化してきました。

現在、一般的なインタフェースはATA、Serial ATA（SATA）、FC、SAS、SCSI、およびiSCSIです。

Advanced Technology Attachment（ATA）

ATAは、マーケティング上の名前としてIDEともいいます。このインタフェースは、x86のPCで一般的に使用されています。このバスは、マスターとスレーブの2つのデバイスをサポートしています。コマンド・キューイングはサポートしていません。このテクノロジーはSATAに急速に置き換わっています。古いドライブではCHSアドレッシング・スキーマが使用され、新しいドライブではLBAが使用されています。

Serial ATA（SATA）

SATAは、ATAディスク形式のシリアル版です。SATAバスにはマスターとスレーブの関係はありません。接続は、小型のケーブルを使ったポイント・ツー・ポイントの接続です。SATAの転送速度は150MB/秒で、第二世代のSATA-IIでは300MB/秒に大幅に向上しています。SATAドライブではLBAスキーマが使用されています。

Small Computer System Interface（SCSI）

SCSIは、ハイエンド・サーバー市場で使用されているテクノロジーで、テープ・ドライブやCD-ROMドライブなどのI/Oデバイスにも使用されています。SCSIでコマンド・タグ・キューイングを使用すると、ATAよりも高いパフォーマンスを実現できます。

Serial Attached SCSI（SAS）

SASは、パラレルSCSIテクノロジーを改良したものです。SASとSCSIの違いは、SASはイニシエータとターゲット間の接続がポイント・ツー・ポイントであり、より高速なデータ転送速度をサポートしている点です。ディスク・アクセスは、引き続きSCSI LBAスキーマに従っています。

ファイバ・チャネル（FC）

ファイバ・チャネルは、シリアルの光インタフェースまたは銅線インタフェースです。シリアルSCSIプロトコルが物理的なFCパケット・フレームのトランスポート層の上に使用されています。FCはおもにハイエンドのストレージ・ソリューションに使用されています。SCSIが使用されているため、SCSI LUNによってLBAのアクセス・スキーマが使用されています。FCを使用するには、専用のストレージ・エリア・ネットワーク（SAN）インフラストラクチャが必要です。

前述のすべてのインタフェース・テクノロジーでバス・テクノロジー転送コマンドの仕様を処理するには、ホスト・バス・アダプタ（HBA）が必要です。

iSCSI

SCSIプロトコルをIPインターコネクト経由で使用するiSCSIは、SANベースのソリューションの低コスト版です。ストレージ・インフラストラクチャを構築するのに特別なハードウェアは必要ありませんが、同じネットワーク・インフラストラクチャを共有する他のシステムによって、パフォーマンスが影響を受ける可能性があります。FCタイプのストレージと同様に、iSCSI LUNにはLBAタイプのアクセス・スキーマを使ってアクセスします。

ストレージのナチュラル・ブロック・サイズ

前述したように、ディスク・テクノロジーで使用されるストレージの最小ユニットはセクターです。従来、セクターのサイズは常に512Bでしたが、ディスク・テクノロジーでデータ・ストレージ密度が増加するにつれて、このサイズでは非効率になっています。さらに大きなデータ・セクターへのニーズが認識されたため、それに対応して、International Disk Drive Equipment and Materials Association（IDEMA）は、アドバンスト・フォーマットと呼ばれるセクターあたり512バイトから4096バイトの標準に移行する、業界全体にわたるイニシアチブを開始しました。このため、ディスクのナチュラル・ブロック・サイズ（NBS）がディスクの物理セクター・サイズになっています。ただし、ディスク・ドライブ・ベンダーの中には、依然として、異なるセクター・サイズを物理的に使用する場合は512Bのブロック・サイズをレポートしているベンダーもあります。そのため、ディスク管理ソフトウェアでこのようなドライブに最適なパーティション・レイアウトを作成するのが難しくなっています。

現在、次の3種類のディスク・ドライブが提供されています。

- 512nディスク－512Bのセクター（512Bのナチュラル・ブロック・サイズ（NBS））を使用するディスク
- 4Knディスク－4096Bのセクターを使用し、4096BのNBSをレポートしているディスク
- 512eディスク － 4096Bのセクター・サイズを使用するが、512BのNBSのセクターをエミュレートまたはレポートしているディスク

これらのディスクはディスク・ラベル付けツールで問題があり、ディスクによってレポートされる情報から実際のNBSを判別できません。

ストレージ・サブシステムからエクスポートされるストレージ・ボリュームについては、NBSに固定値や標準値はありません。LBAのアドレッシング方法がこれらのアプリケーションで使用されるため、LUNのNBSには制限がなく、LUNの作成時に定義されます。ストレージ・サブシステムのソフトウェアが、ストレージ・サブシステム内の物理ディスク・セクターへのLUNブロックのマッピングを管理します。そのため、アプリケーションのI/O特性に最適なブロック・サイズを選択して、アプリケーション・ホストのデータベースまたはファイル・システムのブロック・サイズと同じにできます。ブロック・サイズを選択する際、経験則から、オンライン・トランザクション処理（OLTP）アプリケーションには小さいブロック・サイズ（4～8KB）を選択し、シーケンシャル（順次）やストリーミングのタイプのワークロードには大きいブロック・サイズ（64～128KB）を選択することを推奨します。

ディスク・ドライブへのパーティションの適用

大部分のオペレーティング・システムでは、情報を体系化して格納するため、ディスク・ドライブに何らかの論理ストレージ・ユニットの概念が使用されています。これらの論理ユニットはパーティションとして参照されます。また、スライスとして参照されるオペレーティング・システムもあります。このセクションでは、ストレージ・パーティションの一般的な概念と、主要なオペレーティング・システムでストレージ・パーティションの作成に使用されているさまざまな手法について説明します。

パーティション管理ソフトウェアは、パーティションの作成、削除、サイズ変更、およびパーティション特性の操作に使用します。各オペレーティング・システムには、通常、このようなツールが独自に作成されて含まれています。また、ほとんどの場合、パーティショニング・ソフトウェアによってブート情報とソフトウェアがディスクに追加されます。

パーティション化の標準の変化

マスター・ブート・レコード（MBR）と呼ばれるもっとも広く使用されているパーティション形式は、PC DOSベースのコンピュータ・アーキテクチャに由来しています。このパーティション・スキームは、1つのブート・レコードと最大4つのパーティションで構成されます。MBRはディスクの先頭に配置され、ブート・コードと、4つのパーティション定義が含まれたテーブルが格納されています。各パーティションはパーティションの長さと開始位置で定義され、この長さと開始位置はCHSアドレスと呼ばれるシリンダ、ヘッド、およびセクター（CHS）フィールドで定義されています。セクター・カウントは1（ゼロではなく）で始まり、ヘッド数の上限は255です。シリンダ・フィールドのヘッド上限とサイズ上限から、このスキーマはディスク・サイズが2テラバイト（TB）未満の場合に適しています。

MBRのパーティションではディスク・サイズに上限があるため、新しい標準GUIDパーティション・テーブル（GPT）が開発されました。この標準では、CHSアドレッシングの代わりにLBAを使用してパーティションの位置が定義されます。GPTの定義は、さらに幅広い仕様であるユニファイド・エクステンシブル・ファームウェア・インタフェース（UEFI）の一部となっています。GPTベースのパーティション・スキーマは、多くの場合、EFIシステム・パーティションベースのシステムとみなされます。GPT標準のパーティションでは、CHSのアドレッシングと同様にLBAのカウントはゼロ（1ではなく）で始まります。

ディスク容量が2TBを超える場合は、GPTのパーティション構成が必要です。GPTフォーマットのディスクからオペレーティング・システムが読み取るためには、オペレーティング・システムがGPT fdiskのラベル付け構造をサポートしている必要があります。一部のオペレーティング・システムでは、GPTラベル付きディスクからブートするためには、EFIファームウェア（BIOS）を使用するハードウェアが必要です。

UNIXベースのシステムは、PCのMBRベースのパーティション構成とは異なる独自のディスク・ラベル付け構造で始まりました。このパーティション構成では、ディスク・パーティションにスライスという用語が使用されています。ディスクのジオメトリとスライスのレイアウトに関する情報は、ラベル領域に格納されます。この情報をボリュームの目次（VTOC）といいます。ディスクにVTOC情報を書き込む操作を、多くの場合、ディスクの"ラベル付け"といいます。UNIXがx86プラットフォームに移植されたときに、PC DOSのMBRパーティション・スキーマが引き続き使用されて、MBRパーティション・スキーマの上にUNIXのVTOCスキーマが配置されたため、VTOCパーティション・スキーマはMBRパーティション内に書き込まれるようになっています。

たとえば、Oracle Solarisオペレーティング・システムでは、Oracle Solarisタイプのパーティションは MBRスキーマで作成されます。Oracle Solarisでは、このMBRパーティションはOracle Solarisディスク・ボリュームとして扱われ、このディスク・ボリュームに、ボリュームの定義済みスライスが含まれたUNIX VTOC情報が書き込まれます。このOracle Solarisベースのディスク・ラベルをSMI（Sun Microsystems Inc.）ラベルといいます。

用語"スライス"と"パーティション"を混同して、混乱が生じることが多いです。また、特定のパーティションを指す場合、パーティション階層のどのレベルにパーティションが配置されているのかを明確に示す必要があります。この概念を正しく把握しておかないと、UNIXボリュームが含まれているMBRパーティションの開始位置とUNIXボリューム内のスライスの開始位置の2つのレベルにミスアライメントが生じる可能性があります。

Linuxオペレーティング・システムでは、DOSパーティション・スキーマをそのまま使用しており、追加のスライス・メカニズムは使用していません。Oracle Solarisでは、SMIタイプのパーティション・スキーマで追加のスライス・レベルを使用していますが、新しいEFIラベル付けスキーマはディスクに対してネイティブで、追加のレイヤーは使用しません。

標準の変化がパーティション・ツールとアライメントに及ぼす影響

ディスク・ドライブのストレージ容量とストレージのブロック編成が変化するにつれ、オペレーティング・システムのアドレッシング用にストレージ・デバイスを準備するのに使用するツールが、その変化に対応できなくなっています。

ストレージ・ユニットを提供する今日のストレージ・デバイスとストレージ・サブシステムでは、ハードウェアの元のセクター・サイズ512バイトは使用されなくなっています。新世代のSATAドライブとソリッド・ステート・ドライブ（SSD）では、4KBのNBSが使用されています。Sun ZFS Storage Applianceで提供しているLUNでは、デフォルトで8KBのNBSが用意されています。

ストレージ・ユニットで最適なパフォーマンスを実現するには、ファイル・システムで使用されるブロック構造をストレージ・ユニットのブロック構造とアライメントすることが重要です。ファイル・システムのファイル構造の最初のブロックがストレージ・サブシステムのブロックの先頭で始まっており、ファイル・システムのブロック・サイズがストレージ・デバイスのNBSと同じであるか偶数倍になっていれば、アライメントは適切に行われています。

次の図に、パーティションのミスアライメントが原因で発生した、ファイル・システムのブロックとストレージ・サブシステムのブロックのミスアライメントを示します。この'典型的'なタイプのミスアライメントは、依然としてストレージ・デバイスの512Bサイズのブロックがディスク・パーティション・ツールでデフォルトになっているために発生しています。このパーティション作成モデルはPC DOS環境に由来しており、x86プラットフォームで実行される各種オペレーティング・システムで引き続き使用されています。

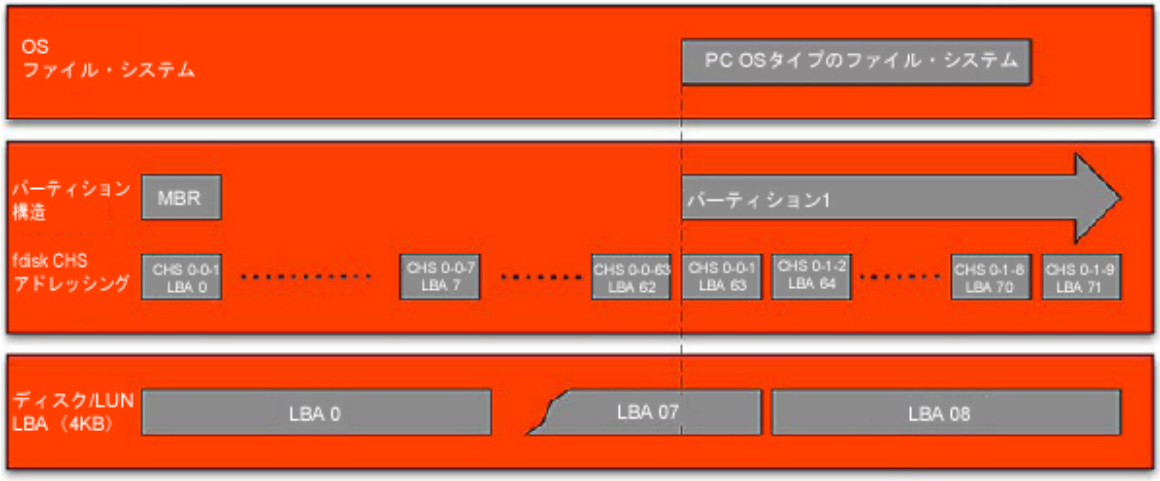

図2. ミスアライメントされたパーティションの例

図2には、ディスクまたはLUNで使用されているLBAブロック構造、パーティション・ツールfdiskからのディスク/LUNのビューとfdiskで作成されたパーティション、および最初のパーティションに作成されたNew Technology File System（NTFS）と呼ばれるMicrosoft Windowsベースのファイル・システム構造の、3つのレイヤーを示しています。

fdiskでは、ブロック構造のタイプおよび基盤となるディスク/LUNのブロック・サイズは認識されません。fdiskでは、セクター63が最初のパーティションの開始セクターに、また512Bがセクターのブロック・サイズにデフォルト設定されます。

新しいオペレーティング・システムとファイル・システムでは、I/Oのミスアライメントを避けるため、この元のパーティション・スキーマは使用されなくなっています。たとえば、Oracle Sun ZFSでは、ディスクをZFSで直接制御する場合は256が開始セクターに使用されます。

次の図に、ディスク/LUNで4KBのNBSを使用した場合を示します。

NTFSベースのファイル・システムのデフォルト・ブロック・サイズ（クラスタ・サイズ）は、ファイル・システムが作成されたパーティションのサイズによって4KB～64KBと異なります。パーティションの開始位置がディスク/LUNのブロックの開始位置とアライメントされないため、NTFSクラスタ・ブロックを読み取るたびに、ディスク/LUNの2つのブロックを読み取る必要があります。ファイル・システムの1つのブロックに書き込むときに、ディスク/LUNの2つのブロックを読み取ってこの2つのブロックに書き込む必要があるため、1つの論理I/Oに4つの物理I/Oが発生します。

fdiskで指定されているパーティションの最初のセクターがディスク/LUNブロックの開始位置に配置されるようにすると、次の図に示すように、ファイル・システム・レイヤーでブロックを更新したときにディスク/LUNに追加のI/Oが発生するのを防ぐことができます。図3に示すように、ファイル・システムのブロックの書込みまたは読取りを行うときに、基盤となるディスク/LUNの1つのブロックにアクセスするだけで済みます。

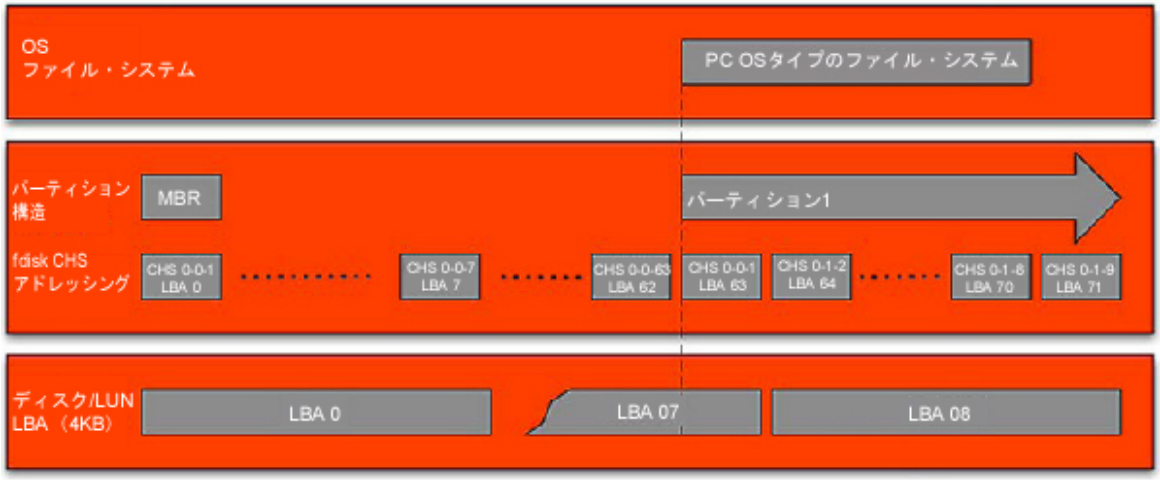

図3. Microsoft Windowsベースのオペレーティング・システムでのfdiskパーティションのアライメントの例

I/Oのミスアライメントの問題は、おもに次の3つの理由から発生します。

- パーティションの作成と管理に提供されているツールで、512Bのセクター・サイズでディスク/LUNがセクターとシリンダに分割される、CHS抽象化モデルがまだなお使用されています。これらのツールでは、LBA抽象化モデルを使用するディスク/LUNは認識されません。
- 大部分のOSプラットフォームでは、ブート情報とディスク・ラベル情報を格納するために、ディスク/LUNの先頭の領域が一部使用されます。この情報が、最初のパーティションの重複部分で上書きされないようにする必要があります。
- 推奨されていませんが、大部分のOSプラットフォームでは、1つのディスク/LUNで複数のパーティションを使用できます。各パーティションは適切にアライメントされる必要があります。

1つのディスク/LUNでの複数のパーティションの使用

1つのディスク/LUNあたり複数のパーティションを使用しても、パーティションのアライメントに特にメリットはありません。可能であれば、アプリケーションまたはファイル・システムでディスク全体を使用するようにすると、パーティションのアライメントを簡素化できます。たとえば、ZFSファイル・システムでは、ディスク全体をZFSプールに追加するように選択できるため、ZFSファイル・システムによってZFSレイヤー内でNBSアライメントが処理されます。

ただし、1つのディスク/LUNあたり複数のパーティションを使用する、さまざまな理由があります。

- ほとんどの場合、コンピュータ・ハードウェアのブート・プロセスで、起動するOSのブート情報へのアクセスにパーティションを使用するように指定されています。
- OSで、インストール時に複数のパーティションを使用する必要があります。
- 複数の異なるOSブート・イメージを同じディスク/LUNに配置します。

- OSのブート・ファイル・システムをアプリケーションのファイル・システムと別にすることが必要になる場合があります。この理由の1つに、バックアップとリストアの設計要件があります。OSのパーティションはアプリケーションのパーティションよりも変更の可能性が少ないため、別々のパーティションを使用すると、それぞれに別々のバックアップ・スケジュールを設定できます。

  また、複数のパーティションを使用すると、マシンを起動するまでのリストア時間を短縮できます。マシンが起動したら、各アプリケーション環境のリカバリ時間目標（RTO）要件の優先順位に応じて、残りのバックアップをリストアできます。

ディスクの準備とパーティション化に関する考慮事項

使用できる選択肢を把握します。ディスク・パーティション・ツールやラベル・ツールを選択できるかどうか、また、インストール・プロセスでこれらのツールを使用できるかどうかを把握します。使用できない場合、パーティション・レイアウトがすでに含まれているディスクを使用できるかどうかを判断します。これらを使用できない場合、Windows 2008などのようにOSによってデフォルトで適切に実行されない限り、アライメントされたパーティションを使用するブート/OS環境は作成できません。

また、ルート・ファイル・システムをアライメントすることが重要かどうかを判断します。ルートとアプリケーション・データを別々のパーティションに配置できる場合は、通常、アプリケーション・データが含まれているパーティションをアライメントするだけで十分です。

インストール時にディスク・ラベル・ツールとパーティション・ツールの使用を選択できる（Oracle Solaris live imageインストール時のpartedツールなど）場合は、セクターごとのアライメントが可能なツールを使用します。つまり、セクター番号を使ってパーティションの開始位置と終了位置を指定できます。

可能な場合、1つのパーティションを使用します。これにより、パーティションの開始位置の計算が単純になり、計算が1回で済みます。1つのパーティションを使用すると、パーティション間の未使用の（無駄な）ディスク領域が減ります。

事前にパーティション化されたドライブをインストール・プロセスで使用できる場合は、OSの標準のブート・イメージを使い慣れたツールで使用し、要件に従ってドライブをパーティション化してから、OSのインストール・プロセスを開始します。x86プラットフォーム用のスイス・アーミー・ナイフのような優れたツール・キットがhttp://www.ultimatebootcd.com/とhttp:/www.sysresccd.org/Main_Pageで紹介されています。

ブート・ディスクで、セクターの最初の領域がラベル情報とブート情報に予約されることに注意してください。通常は最初のパーティションにセクター64を使用すれば間違いありませんが、お使いの製品の情報を確認してください。

OS環境の実行を開始し、新しいディスク/LUNの追加が必要になったら、GNU **parted**ツールなどのセクター・モードで動作するパーティション・ツールを使用します。

ファイル・システムのブロックを基盤となるディスク/LUNのブロック構造と確実にアライメントするため、ファイル・システムのブロック・サイズは、ディスク/LUNのNBSと同じにするか、ディスク/LUNのNBSのサイズの2のべき乗にします。Sun ZFS Storage Applianceで提供しているLUNのように、LUNの作成時にディスク/LUNのブロック・サイズを定義できる場合もあります。OLTPタイプのワークロードでは、LUNのブロック・サイズは、ディスク/LUNを使用するファイル・システムやアプリケーションで使用されているブロック・サイズと同じにします。

直接接続型ディスクと比較したストレージ・サブシステムのLUNのアライメント

前述したように、ストレージ・サブシステムまたは直接接続型のディスクを使用して、オペレーティング・システムの割当てのディスク領域を指定できます。直接接続型のディスクを処理する場合、OSは、ディスクのファームウェアおよびディスクのジオメトリ特性の表示と直接インタフェースを取ります。

ただし、ストレージ・サブシステムのLUNを使用する場合は、ディスク障害からの保護を提供する、RAIDタイプのデータ保護機能を備えたストレージ・サブシステム・ソフトウェアによって使用される、追加の抽象化レイヤーが存在します。アレイから分配された容量のLUNが、複数の物理ディスクに分散されます。アレイ・ソフトウェアによって、LUNの論理ブロックと基盤となる物理ディスク・ブロックの間でアライメントが適切に行われます。Sun ZFS Storage Applianceの場合、Oracle SolarisとSun ZFS Storage Applianceが内蔵ディスク・ストレージと容量の管理を行って、ストレージ・ソフトウェア・スタックの各レイヤー間でブロックが適切にアライメントされるようにします。

注：以降のセクションに示されているSun ZFSとOracle Solarisの個別の参考資料では、Oracle Solaris OSを実行しZFSファイル・システムを使用するクライアントとSun ZFS Storage ApplianceのLUNを使用するクライアントについて述べています。これらの参考資料は、Sun ZFS Storage Appliance自体には適用されません。

Sun ZFS Storage ApplianceでのLUNの使用準備

Sun ZFS Storage Applianceでは、ファイバ・チャネル・プロトコルまたはiSCSIプロトコルを使用してブロックのストレージ・ボリュームをクライアントで使用可能にできます。

いずれの場合も、Sun ZFS Storage Applianceのブラウザ・ユーザー・インタフェース（BUI）に用意されている次のダイアログ・ウィンドウを使用して、ストレージ・プールからボリュームを作成します。

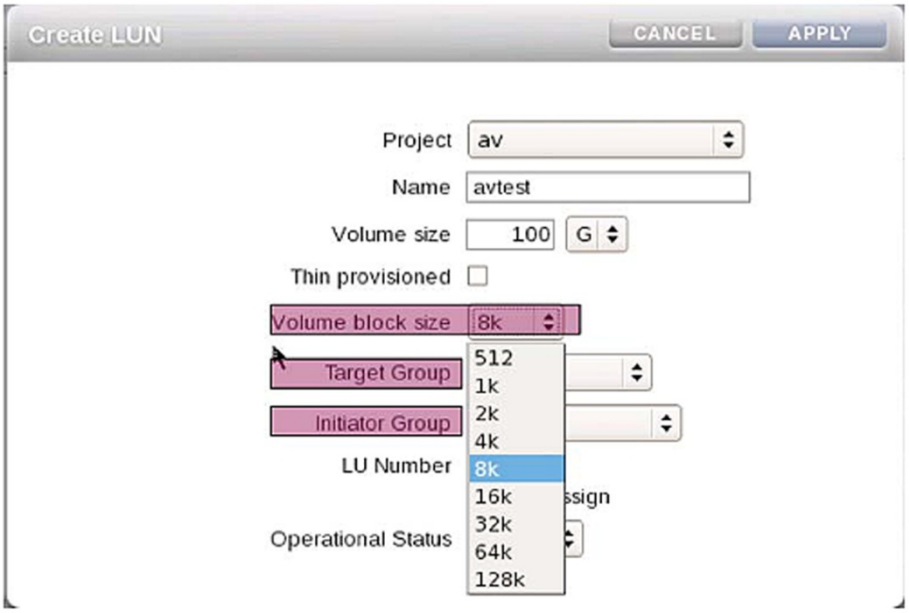

図4. Sun ZFS Storage ApplianceでのLUNの作成

LUNをいったん作成するとLUNのブロック・サイズは変更できないため、LUNのブロック・サイズは慎重に選択してください。ブロック・サイズが、LUNを使用するアプリケーションのパフォーマンス要件に対応している必要があります。

ディスク・パーティションの使用を選択するときに、前述したLUNパーティション・アライメントの注意事項を考慮してください。

アクセス制限を定義する処理の一環として、LUNがFCターゲット・グループまたはiSCSIターゲット・グループおよびイニシエータ・グループに割り当てられます。イニシエータ・グループのメンバーであるホスト（HBA）のみ、このLUNにアクセスできます。

FC環境でのSun ZFS Storage Appliance

Sun ZFS Storage ApplianceのLUNを使用する場合、1つのLUNに複数の物理パスを使用します。FC接続では、次の図に示すように、2つの独立したSANファブリックのトポロジと1つのSun ZFS Storage Applianceクラスタ構成を使用することを推奨しています。

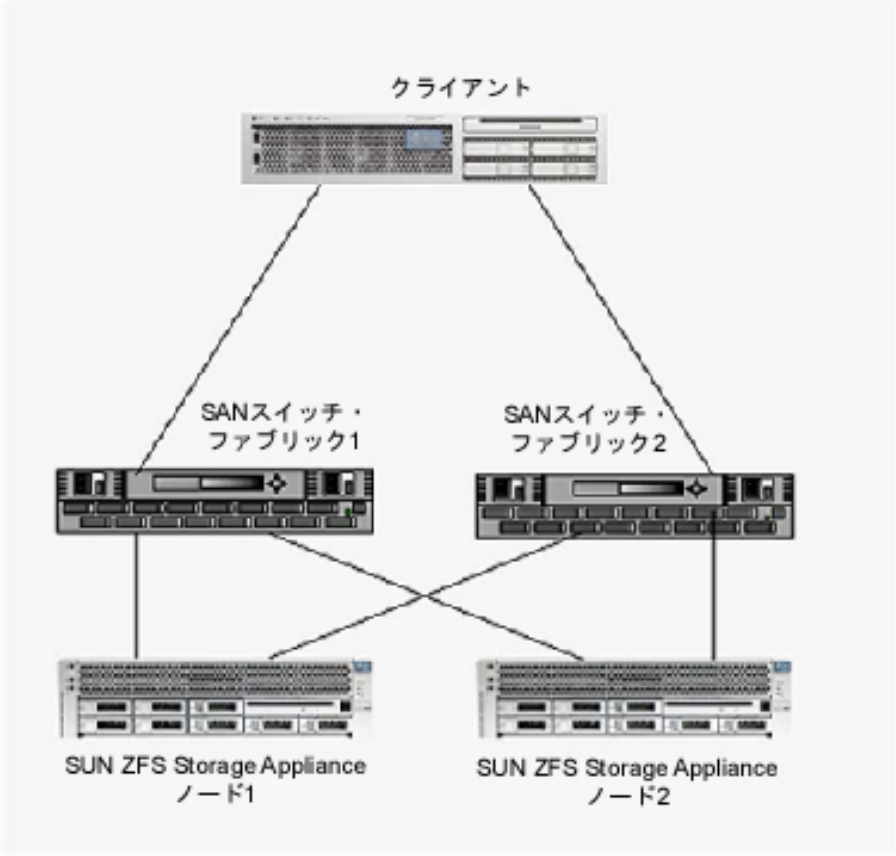

図5. Sun ZFS Storage Applianceの冗長SAN構成

この最適な冗長SAN構成で、ファイバ・チャネル・パスのコンポーネント障害から保護できます。このような障害により、Sun ZFS Storage Applianceのノード内のLUNでパス障害が引き起こされます。

Sun ZFS Storage Applianceのノード障害が発生した場合、Sun ZFS Storage Applianceの残りのノードによって、障害が発生したノードのストレージの制御が再開されるとともに、稼働中のノードのLUNが使用されます。この構成では、クライアントFC HBAが相互に参照しないように、ファブリックに適切なゾーニングを設定し、Sun ZFS Storage Applianceのノードに適切なターゲット・グループとイニシエータ・グループを設定する必要があります。Sun ZFS Storage Applianceでのファイバ・チャネルの使用の詳細については、『Understanding the Use of Fibre Channel in the Sun ZFS Storage Appliance』を参照してください（ドキュメントの場所については、付録を参照）。

iSCSI環境でのSun ZFS Storage Appliance

iSCSIプロトコルを使ってホスト環境にLUNを提供する場合、Sun ZFS Storage ApplianceのIPマルチパス（IPMP）機能を使用してネットワーク・インフラストラクチャにパスの冗長性を構築できます。

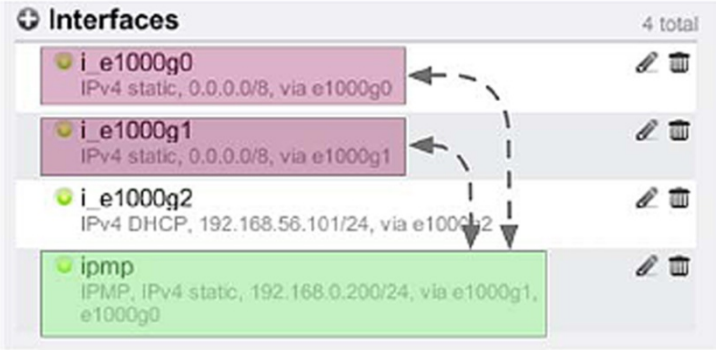

図6. Sun ZFS Storage ApplianceのBUIに表示されたLAN IPMPグループ

IPMPグループには、グループ内のアクティブな物理ポートに割り当てられている論理IPアドレスが含まれます。このポートを経由したネットワーク接続が失われると、IPトラフィックとこのポートのIPアドレスがグループ内の別のポートに自動的にルーティングされます。他のIPポートが同じサブネット（LAN、InfiniBand[IB]パーティション、またはVLAN）に配置されている場合は、任意の番号のIPポートをIPMPグループに割当てできます。

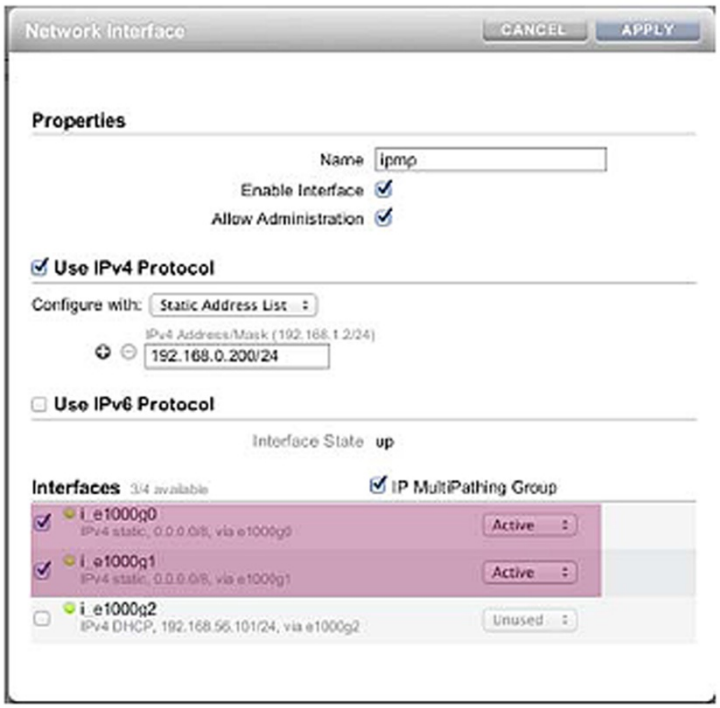

図7. Sun ZFS Storage ApplianceのBUIに表示されたLAN IPMPグループの設定

IPMPの使用の詳細については、Sun ZFS Storage Applianceのドキュメント（付録を参照）を参照してください。LUNを作成する前に、iSCSIイニシエータ・グループとターゲット・グループが適切に構成されていることを確認します。

Oracle Solarisでのディスク管理

Oracle Solarisオペレーティング・システムは、SPARCプロセッサベースとx86プロセッサベースの両方のシステムでサポートされています。このオペレーティング・システムでは、ディスクにラベルを付けてディスク/LUNにスライスを作成するために、formatユーティリティを使用しています。前述したように、Oracle Solarisでは次の2種類のディスク・ラベルをサポートしています。

- SMI − Oracle Solarisの従来のボリュームの目次（VTOC）ラベル。x86プラットフォームでは、SMIディスク・ラベルはMBR fdiskパーティション内に作成されます。Oracle Solaris x86ベースのシステムでOracle Solarisをディスクからブートするためには、まず、fdiskタイプのユーティリティを使用してディスクをパーティション化する必要があります。Oracle Solarisのインストール・プロセスでは、fdiskタイプのMBRパーティションが作成されてから、MBRタイプのパーティション内のOracle Solaris SMIスライスにOracle Solarisのルート・ファイル・システムがインストールされます。
- EFI − fdisk/SMIラベル構造での2TBのサイズ上限を解消する、新しいディスク・ラベル。ディスクが2TBを超える場合は、64ビットのOracle Solarisカーネルが必要です。EFIはGPTパーティションともいいます。

どちらのディスク・ラベル・スキーマが個々の状況に最適なのかを判断するには、以下の特性とEFIの制限について検討します。

- EFIディスク・ラベル・スキーマでは、サイズが2TBを超えるディスクをサポートしています。
- EFIディスク・ラベル・スキーマでは、使用可能なスライスは0〜6で、スライス2は通常のスライスです。
- パーティション（またはスライス）を一次ラベルやバックアップ・ラベル、または他のスライスと重複させることはできません。通常、EFIラベルのサイズは34セクターであるため、スライスはセクター34で開始されます。つまり、セクター0（ゼロ）で開始されるパーティションは存在しません。
- EFIラベルにはシリンダ、ヘッド、およびセクターの情報はいっさい格納されません。サイズはブロック単位でレポートされます。
- これまで代替シリンダ領域に格納されていた情報は、ディスクの最後の2つのシリンダまたはOracle Solarisパーティションに格納されます。
- formatユーティリティを使ってパーティション・サイズを変更する場合、サイズがゼロのパーティションにはunassignedパーティション・タグが割り当てられます。デフォルトでは、formatユーティリティによって、ゼロより大きいサイズのパーティションにはusrパーティション・タグが割り当てられます。パーティションを変更したあとでパーティション・タグを再度割り当てるには、パーティション変更メニューを使用します。ただし、サイズがゼロでないパーティションをunassignedパーティション・タグに変更することはできません。
- VTOCラベル付きディスクを使用するシステム向けの階層化されたソフトウェア製品で、EFIディスク・ラベル付きディスクにアクセスできないことがあります。
- Oracle Solaris 11.1以降では、EFIディスク・ラベル付きディスクからのブートをサポートしています。EFIラベル付きディスクをサポートしているSPARCシステムではブート・ファームウェアが必要です。また、x86プラットフォームではGrand Unified Bootloader（GRUB）2.1環境が必要です。
- EFIディスク・ラベルは、IDEディスクではサポートされていません。

- サイズが2TBを超えるディスクにアクセスするには、64ビットのOracle Solarisカーネルが必要です。
- x86ベースのシステムでは、EFIラベル付きのサイズが2TBを超えるディスクでOracle Solarisのfdiskコマンドを使用できます。
- EFI仕様では、スライスの重複は禁止されています。ディスク全体はcxtydzで表されます。スライスはcxtydzsnで表されます。ここで、nはスライス番号です。
- EFIディスク・ラベルでは、ディスクやパーティションのサイズ情報がセクター単位およびブロック単位で提供されます。シリンダ単位およびヘッド単位では提供されません。
- 次のformatユーティリティ・オプションは、EFIラベル付きディスクではサポートされていないか適用できません。
  - EFIラベル付きディスクではformat.datファイルにエントリが必要ないため、`save`オプションはサポートされません。
  - `backup`オプションは適用できません。

Oracle Solaris 11.1以降では、EFIが、EFIラベル付きディスクからのブートがサポートされているプラットフォームでインストール・プロセスに使用されるデフォルトのディスク・ラベルです。x86ベースのシステムでは、EFIラベル付きディスクのパーティションからのOracle Solarisのブートをサポートするため、ブート管理ユーティリティGRUBが適応されています。また、I/OスタックとOracle Solarisのディスク・ユーティリティを通じて、ディスク/LUNのNBSを検出し、ディスク/LUNのNBSのサイズに適切にアライメントされたパーティションを作成できるようになりました。すべてのストレージ・ベンダーがNBSを正しくレポートしているわけではありません。そのような場合は、以降の例に示すように、ユーティリティfdiskとformatで標準のデフォルトが使用されます。

これらすべては、Oracle Solarisを使用する場合のパーティション・アライメントで何を意味するのでしょうか。このホワイト・ペーパーでは、iSCSIまたはファイバ・チャネルを使用するSun ZFS Storage ApplianceからOracle Solarisに提供されるLUNについて説明しています。ほとんどの場合、Oracle Solarisのシステム・パーティションはローカル・ディスクに配置されます。通常、これらのディスクでは、アライメントの問題が当てはまらない512Bのセクター・サイズが引き続き使用されています。SSDをブート・デバイスとして使用する場合など、パーティションのアライメントが問題となる状況では、Oracle Solarisのルート・ファイル・システムに使用するディスクに、以下の方法でパーティションとスライスのアライメントを行うこともできます。Oracle Solarisのインストール・プロセスに精通しており、どこで標準のインストール手順を中断してパーティションおよびOracle Solarisのスライスのレイアウトをカスタマイズするのかを把握している必要があります。ただし、このトピックは、このホワイト・ペーパーの対象範囲外となります。

アプリケーションとアプリケーション・データセットのストレージに別々のディスクを使用できる場合、これを選択する必要があります。以降のセクションでは、SMIとEFIの両方のディスク・ラベル・スキーマについて、x86プラットフォームとSPARCプラットフォームにおけるパーティションの問題とOracle Solarisのスライスの問題について説明します。

x86ベースのシステムでのOracle Solaris

PCタイプのアーキテクチャのディスク・パーティション構造は、そのアーキテクチャで使用されているBIOSファームウェアに依存します。電源投入フェーズではBIOSがディスクI/Oを制御するため、マスター・ブート・レコード（MBR）をロードできるパーティション構造およびOSのブート開始ソフトウェアをロードできるMBRをBIOSが認識する必要があります。そのため、MBRのDOSタイプのパーティション構造がブート可能ディスクに配置されている必要があります。

Oracle Solarisでは独自のディスク・ラベル構造を使用しています。Oracle Solaris 11.1以降では、GNU GRUBバージョン2.1によりEFI（GPT）ラベル付きディスクからのブートをサポートしています。Oracle Solarisのformatユーティリティでは、MBRタイプのパーティション内にスライス構造が作成されます。

SMIディスク・ラベルの使用

SMIディスク・ラベルを使用する場合、FDISK/MBRパーティションの開始位置とSMIスライスの開始位置の2つのレベルのアライメントを考慮します。Oracle SolarisのfdiskユーティリティとformatユーティリティのいずれもCHSアドレッシング・モードで機能します。これらのいずれのツールでも、シリンダ1が開始シリンダとして使用されて、シリンダ0がブートとOSでの使用に予約されます。formatユーティリティでは、FDISKパーティションの開始位置からシリンダの番号付けが始まります。複数のFDISKパーティションを使用する場合、SMIの各VTOCはFDISKパーティション内のシリンダ0で始まります。

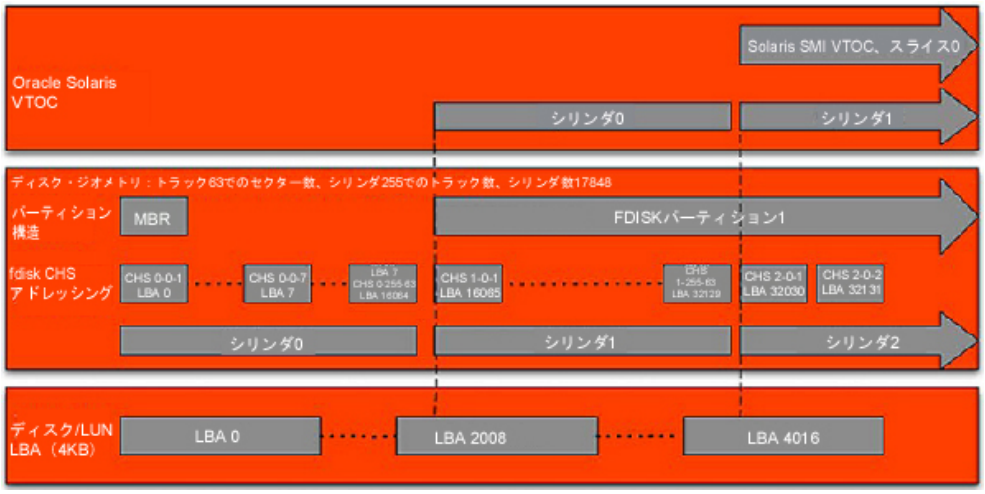

図8. x86システムでOracle Solarisを使用する場合の、fdiskとSMI VTOCのデフォルト・レイアウト

Oracle Solarisのfdiskコマンドと`format`コマンドを使用して、この図に示されているレイアウトを確認します。Oracle Solarisの`fdisk`コマンド`-W - <rawdisk>`を使用して、ディスク・パーティションのレイアウトを確認します。

```
root@aie-linlithgow:~#fdisk -W - /dev/rdsk/c6t0d0p0
* /dev/rdsk/c6t0d0p0 default fdisk table
* Dimensions:
*    512 bytes/sector
*     63 sectors/track *    255 tracks/cylinder
*   17848 cylinders
*
* systid:

*    1: DOSOS12
*    2: PCIXOS
*    4: DOSOS16
…
*  191: SUNIXOS2
*  238: EFI_PMBR
*  239: EFI_FS
*
*Id    Act  Bhead  Bsect  Bcyl    Ehead  Esect  Ecyl    Rsect      Numsect
 191   128  0      1      1       254    63     1023    16065      286712055
 0     0    0      0      0       0      0      0       0          0
 0     0    0      0      0       0      0      0       0          0
 0     0    0      0      0       0      0      0       0          0
```

このテーブルから、ディスクc6t0d0にCHS 10-1、Bcyl = 1、Bhead=0、Bsect=1で始まるパーティションが1つ作成されていることが分かります。

Oracle Solarisのformatコマンドを使用して、FDISKパーティション内のOracle Solaris SMIスライス構造のレイアウトを確認できます。

```
root@aie-linlithgow:~# format /dev/rdsk/c6t0d0s2
selecting /dev/rdsk/c6t0d0s2
[disk formatted]
/dev/dsk/c6t0d0s0 is part of active ZFS pool rpool. Please see zpool(1M).
/dev/dsk/c6t0d0s3 is part of active ZFS pool homes. Please see zpool(1M).

FORMAT MENU:
        disk       - select a disk
        …
        quit
format> partition
PARTITION MENU:
        0      - change `0' partition
        …
        print  - display the current table
        …
partition> print
Current partition table (original):
Total disk cylinders available: 17845 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       1 -  5312       40.69GB    (5312/0/0)   85337280
  1 unassigned    wm       0                0         (0/0/0)             0
  2     backup    wu       0 - 17844      136.70GB    (17845/0/0) 286679925
  3        usr    wm    5313 - 17844       96.00GB    (12532/0/0) 201326580
  4 unassigned    wm       0                0         (0/0/0)             0
  5 unassigned    wm       0                0         (0/0/0)             0
  6 unassigned    wm       0                0         (0/0/0)             0
  7 unassigned    wm       0                0         (0/0/0)             0
  8       boot    wu       0 -     0        7.84MB    (1/0/0)         16065
  9 unassigned    wm       0                0         (0/0/0)             0

partition> quit
```

トラックあたりのセクター数とシリンダあたりのトラック数は変更できません。そのため、SMIスライスの開始位置とLUNブロックをアライメントするのは難しいです。スライスをLUNのディスク・ブロック構造とアライメントするには、次が当てはまる必要があります。

(partition start cylinder + slice start cylinder) * sectors per cylinder * sector size) % LUN blocksize == 0

Oracle Soarisのデフォルトのパーティション・レイアウトとスライス・レイアウトを使用したこの例では、LUN LBAのさまざまなブロック・サイズについてスライス開始シリンダを計算すると、次のようになります。

表1. さまざまなNBS値でのSMIパーティションのアライメント

SUN ZFS STORAGE APPLIANCEのLUNのブロック・サイズ	パーティション開始シリンダ	スライス開始シリンダ	実際の開始シリンダ
4 KB	1	7	8
8 KB	1	15	16
16 KB	1	31	32
32 KB	1	63	64
64 KB	1	127	128
128 KB	1	255	256

EFIディスク・ラベルの使用

Oracle SolarisのEFIタイプ・ディスク・ラベルは、format −eコマンドで作成します。このコマンドで、ディスク全体を使用するスライス構造が作成されます。スライスの開始位置と終了位置は、セクター0で始まるサイズが512Bのセクター単位で表されます。最初のスライスのデフォルトの開始位置はセクター34です。次の図に、スライス0およびスライス0の上に配置されたファイル・システムは、基盤となるLUNボリュームのブロックの開始位置とアライメントされないことを示します。

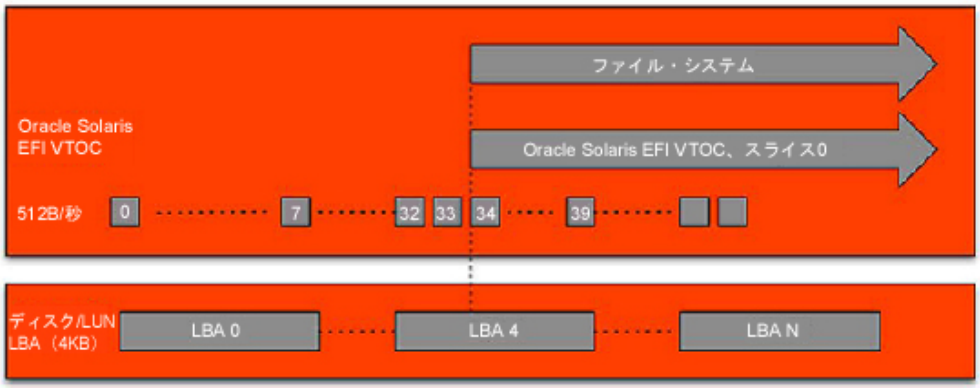

図9. Oracle SolarisでのデフォルトのEFIパーティション・レイアウト

ZFSファイル・システムの4Kのブロック・サイズを使用して、LUNの各I/OがLUNボリュームの2つのブロックに分散されます。format −eコマンドのpartitionオプションで、スライスの開始セクターを変更するオプションがオフセットされます。オフセット値256を使用して、スライスがLUNのブロック・サイズの2のべき乗（最大128K）でアライメントされます。

警告：これは**相殺的な**処理です。レイアウトを変更するパーティションに格納されているデータはすべて**失われます**。

zpoolユーティリティを使用してディスクをZFSプールに直接追加する場合、このユーティリティによってセクター256で始まるパーティションが作成されます。これにより、作成されたパーティションの上にZFSファイル・システム用に適切にアライメントされたパーティションが自動的に作成されます。

```
ymachine: ~# format -e
format>label
[0] SMI Label
[1] EFI Label
Specify Label type[1]: 1
Ready to label disk, continue? Y

format> partition
       ~~~~~~~~~~~~~~
partition> print
Current partition table (original):
Total disk sectors available: 209698782 + 16384 (reserved sectors)

Part      Tag    Flag     First Sector         Size            Last Sector
  0        usr    wm            34            99.00GB          207618081
  1 unassigned    wm             0                0                   0
  2 unassigned    wm             0                0                   0
  3 unassigned    wm             0                0                   0
  4 unassigned    wm             0                0                   0
  5 unassigned    wm             0                0                   0
  6 unassigned    wm             0                0                   0
  7 unassigned    wm             0                0                   0
  8   reserved    wm     209698783             8.00MB          209715166

partition> 0
Part      Tag    Flag     First Sector         Size         Last Sector
  0        usr    wm                34       99.00GB          207618081
Enter partition id tag[usr]:
Enter partition permission flags[wm]:
Enter new starting Sector[34]: 256
Enter partition size[207618048b, 207618303e, 101376mb, 99gb, 0tb]:209698782e
partition> print
Current partition table (unnamed):
Total disk sectors available: 209698782 + 16384 (reserved sectors)

Part      Tag    Flag     First Sector      Size            Last Sector
  0        usr    wm         256           99.00GB          207618303
  1 unassigned    wm           0              0                   0
  2 unassigned    wm           0              0                   0
  3 unassigned    wm           0              0                   0
  4 unassigned    wm           0              0                   0
  5 unassigned    wm           0              0                   0
  6 unassigned    wm           0              0                   0
  7 unassigned    wm           0              0                   0
  8   reserved    wm   209698783            8.00MB          209715166
partition>label
0] SMI Label
[1] EFI Label
Specify Label type[1]: 1
Ready to label disk, continue? Y
partition> quit
label>
Mymachine: ~#
```

`zpool create/add`コマンドを使用すると、完全なLUNがデフォルトの開始セクター256で使用されます。そのため、`zpool create/add`を使って作成されたEFIスライスは常に、LUNの2のべき乗（最大128KB）で適切にアライメントされます。

x86プラットフォームでこの方法でラベル付けされたLUNによってLUNのMBRタイプのラベル（存在する場合）が置き換えられてから、MBRタイプのすべてのパーティションとデータが失われます。

SPARCでのOracle Solaris

SPARC上のOracle Solarisにラベル付けされたディスクには、ディスクに直接書き込まれるスライス構造があるため、x86ベースのプラットフォームのようなMBRパーティション・アライメントの問題は生じません。現在、SMIディスク・ラベル・スキーマのみブート・ディスクでサポートされています。

SMIディスク・ラベルの使用

前述したように、SMIラベル付けスキーマではCHSアドレッシング方法を使用しています。つまり、スライスは常にシリンダ境界で始まります。SMIラベルでラベル付けされたディスク/LUNとデフォルトのスライス・レイアウトは、シリンダ0で始まります。つまり、スライス0は常に適切にアライメントされます。デフォルトのスライス・レイアウトによって、ルート・スライスにわずかなスペースが作成され、スペースがスライス1（swap）とスライス6（usr）に割り当てられます。余分なスライスは、基盤となるLUNブロック構造とはアライメントされません。

```
format> verify
Primary label contents:

Volume name = <        >
ascii name  = <SUN-Sun Storage 7210-1.0 cyl 323 alt 2 hd 254 sec 254>
pcyl        =  325
ncyl        =  323
acyl        =    2
nhead       =  254
nsect       =  254
Part      Tag    Flag     Cylinders       Size            Blocks
 0       root    wm       0 -   4      157.51MB    (5/0/0)     322580
 1       swap    wu       5 -   9      157.51MB    (5/0/0)     322580
 2     backup    wu       0 - 322        9.94GB    (323/0/0) 20838668
 3 unassigned    wm       0              0         (0/0/0)          0
 4 unassigned    wm       0              0         (0/0/0)          0
 5 unassigned    wm       0              0         (0/0/0)          0
 6        usr    wm      10 - 322        9.63GB    (313/0/0) 20193508
 7 unassigned    wm       0              0         (0/0/0)          0

format>
```

この情報からnsect（number of sectors per cylinder：シリンダあたりのセクター数）を使用し、次の計算式を適用して、スライスの開始位置が基盤となるLUN NBSにアライメントされているかどうかを確認できます。

(cylinder number * sectors/cylinder * sector size) % LUN blocksize == 0

ミスアライメントの問題を回避するため、すべてのディスク領域が含まれた1つの大きいスライスを作成することを推奨します。スライス0をfree hogパーティションとして使用し、未割当てのすべてのディスク領域がここに集められるようにします。

```
format> partition

PARTITION MENU:


        0      - change `0' partition
~~~~~~~~~
 !<cmd> - execute <cmd>, then return
 quit
partition> modify
Select partitioning base:
 0. Current partition table (original)
 1. All Free Hog
Choose base (enter number) [0]? 1

Part      Tag    Flag     Cylinders       Size            Blocks
 0       root    wm       0               0         (0/0/0)          0
 1       swap    wu       0               0         (0/0/0)          0
 2     backup    wu       0 - 322         9.94GB    (323/0/0) 20838668
 3 unassigned    wm       0               0         (0/0/0)          0
 4 unassigned    wm       0               0         (0/0/0)          0
 5 unassigned    wm       0               0         (0/0/0)          0
 6        usr    wm       0               0         (0/0/0)          0
 7 unassigned    wm       0               0         (0/0/0)          0

Do you wish to continue creating a new partition
table based on above table[yes]?
Free Hog partition[6]? 0
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:

Part      Tag    Flag     Cylinders       Size            Blocks
 0       root    wm       0 - 322         9.94GB    (323/0/0) 20838668
 1       swap    wu       0               0         (0/0/0)          0
 2     backup    wu       0 - 322         9.94GB    (323/0/0) 20838668
 3 unassigned    wm       0               0         (0/0/0)          0
 4 unassigned    wm       0               0         (0/0/0)          0
 5 unassigned    wm       0               0         (0/0/0)          0
 6        usr    wm       0               0         (0/0/0)          0
 7 unassigned    wm       0               0         (0/0/0)          0

Okay to make this the current partition table[yes]?
Enter table name (remember quotes): "c0t600144F0F05E906C00004F200EA50001d0"

Ready to label disk, continue? Yes

partition>
```

EFIディスク・ラベルの使用

Oracle Solaris SPARCで使用されているEFIディスク・ラベル付けスキーマはx86システムで実行されるOracle Solarisのスキーマと同じであるため、`format -e`コマンドを使用してディスク/LUNにラベル付けすると（セクション"x86ベースのシステムでのOracle Solaris"の手順に従って）、同じデフォルトの開始セクター34が生成されます。スライス0の開始セクターを変更するには、`format -e`コマンドを使用します。最初の34のセクターにはEFIディスク・ラベル情報が含まれるため、開始セクターはセクター34以降にする必要があります。

コマンド`zpool`を使用してディスクをプールに追加すると、ZFSによってスライス0に開始セクター256が使用されて、ディスク全体が使用されます。

Oracle Solarisのパーティション・アライメントの概要

次の表に、Oracle Solarisのクライアントでディスクにラベル付けする場合のさまざまなデフォルトの概要を示します。

表2. Oracle Solarisのパーティションのデフォルトの概要

クライアントOS	SMI バージョン11.1より前のOracle SolarisのCHSアドレッシングのブート・ディスク/LUNに使用。	EFI Oracle Solaris 11.1のブート・ディスク/LUNおよびセクター0で始まるデータ・ディスク/LUN LBA（セクター）のアドレッシングに使用。
x86でのOracle Solaris	PC fdiskパーティションの上に配置。 fdiskパーティションはシリンダ1で始まる。 スライス0が作成されるfdiskパーティションの開始位置に対応して、スライス0はシリンダ1で始まる。	format -eコマンドでは、セクター34がスライス0のデフォルトの開始位置として使用される。 ZFS zpoolでは、セクター256が開始位置として使用される。
SPARCでのOracle Solaris	ディスクに直接配置され、シリンダ0で始まる。	

Oracle Solarisのクライアントに関する推奨事項

可能な場合は、データ・ディスクにEFIタイプのラベルを使用します。`format -e`コマンドを使ってディスク/LUNに手動でラベル付けする場合は、開始セクター256を修正します。これにより、最大128KBのブロック・サイズで、Sun ZFS Storage ApplianceのLUNの基盤となるブロック構造と確実にアライメントできます。開始セクターの値に、ディスク/LUNをパーティション化するときに使用する領域オフセットの業界標準として現在受け入れられている2048（1MB）を使用します。

Oracle SolarisのX86クライアントでSMIタイプのディスク・ラベルが必要な場合、Oracle Solarisのルート・ファイル・システム全体を保持するスライスを1つ作成します。このスライスの開始シリンダについて、次の計算式を使用して正しい開始シリンダ値を取得します。

(cylinder number * sectors/cylinder * sector size) % LUN blocksize == 0

Oracle SolarisホストOSにおけるSun ZFS Storage ApplianceのLUN構成のヒント

マルチパス環境でFC LUNを使用する場合、`stmsboot -e`コマンドを使用してOracle Solarisカーネル構成でこのマルチパスを有効にします。

次のコマンドを使用して、ホストHBAおよび検出されたLUNを表示します。

```
root@aie-perth:~# prtconf -vp | fgrep -i wwn
 node-wwn:2000001b.3284b467
 port-wwn:2100001b.3284b467
 node-wwn:2001001b.32a4b467
 port-wwn:2101001b.32a4b467
 node-wwn:2000001b.32859020
 port-wwn:2100001b.32859020
 node-wwn:2001001b.32a59020
 port-wwn:2101001b.32a59020
root@aie-perth:~#
```

luxadm probeコマンドを使用してLUNを特定します。

```
root@aie-perth:~# luxadm probe
No Network Array enclosures found in /dev/es

Found Fibre Channel device(s):
Node WWN:200100e08bb2a1cf Device Type:Disk device
Logical Path:/dev/rdsk/c0t600144F0F05E906C00004F200EA50001d0s2
root@aie-perth:~#
```

検出されたLUNの表示された各デバイス・パスで、そのLUNに使用されている物理パスの合計数や、SANへの接続に使用されるシステムHBAポートなどの、LUNの詳細情報を確認できます。

```
root@aie-perth:~# luxadm display /dev/rdsk/c0t600144F0F05E906C00004F200EA50001d0s2
DEVICE PROPERTIES for disk: /dev/rdsk/c0t600144F0F05E906C00004F200EA50001d0s2
  Vendor:               SUN
  Product ID:           Sun Storage 7210
  Revision:             1.0
  Serial Num:
  Unformatted capacity: 10240.000 Mbytes
  Read Cache:           Enabled
    Minimum prefetch:   0x0
    Maximum prefetch:   0x0
  Device Type:          Disk device
  Path(s):

  /dev/rdsk/c0t600144F0F05E906C00004F200EA50001d0s2
  /devices/scsi_vhci/ssd@g600144f0f05e906c00004f200ea50001:c,raw
   Controller           /devices/pci@0/pci@0/pci@8/pci@0/pci@a/SUNW,qlc@0/fp@0,0
    Device Address              210100e08bb2a1cf,18
    Host controller port WWN    2100001b32859020
    Class                       primary
    State                       ONLINE
   Controller           /devices/pci@0/pci@0/pci@8/pci@0/pci@a/SUNW,qlc@0,1/fp@0,0
    Device Address              210000e08b92a1cf,18
    Host controller port WWN    2101001b32a59020
    Class                       primary
    State                       ONLINE

root@aie-perth:~#
```

Linuxでのディスク管理

大部分のLinuxディストリビューションはx86ベースのプラットフォームで動作するため、これらのディストリビューションでは'標準'のPCタイプのディスク・パーティション・メカニズムが使用されています。x86以外のハードウェア・プラットフォーム用のディスク・パーティション・ツールについては、ここでは説明していません。Linuxディストリビューションには常に、ディスクをパーティション化するためのfdiskタイプのパーティション・ツールが含まれています。一部のディストリビューションにはGNU partedツールも含まれています。Linuxディストリビューションには、ファイル・システムと基盤となるLUNブロック構造にミスアライメントがないかどうかディスクI/Oを分析するツールは含まれていません。

Linuxのfdiskパーティション・ツールでも、シリンダを使用してパーティションの開始位置と終了位置が定義されます。シリンダの開始位置が、基盤となるZFS LUNファイル・システムのブロックとアライメントされる必要があります。Linuxの`fdisk`コマンドには、トラックあたりのセクター数を指定する`-S`オプションがあります。ZFSブロック・サイズの倍数の値を使用して、アライメントを確実に実行できます。

ただし、実行する際に問題があります。`fdisk -S`オプションでは、63を超える値は受け付けていません。多くの場合、63がデフォルト値として使用されています。そのため、ZFSのブロック・サイズが1KBを超える場合は常にミスアライメントが発生します。

32を指定し、`fdisk -u`オプションを使用してパーティションの開始値をセクター単位で指定することを推奨します。ZFSのすべてのブロック・サイズに対応するには、最初のパーティションに256を指定し、以降のすべてのパーティションに256の倍数の値を指定します。256の倍数の値を使用し、`+n`オプションを使ってパーティション内のセクター数を指定します。ここで、`n`はパーティション内のセクター数です。

```
root@petertest /root % fdisk -u -S 32 /dev/sda
 The number of cylinders for this disk is set to 5140

Command (m for help): n
Command action
   e   extended
 p   primary partition (1-4)p
Partition number (1-4): 1
First sector (32-41942399, default 32):256
Using default value 256
Last sector, +sectors or +size{K,M,G} (256-41942399, default 41942399):
Using default value   419424399
 Command (m for help): p
 Disk /dev/sda: 21.4 GB, 21474836480 bytes
255 heads, 32 sectors/track, 5140 cylinders, total  41942400 sectors
Units = sectors of1* 512 bytes
Disk identifier: 0x000da626

 Device   Boot      Start       End      Blocks   Id  System
/dev/sda1   *         256      41942399    106080   83  Linux
 Command (m for help): w
 The partition table has been altered! Calling ioctl() to re-read partition table.
 Syncing disks.
root@petertest /root % fdisk -l -u /dev/sda
 Disk /dev/sda: 21.4 GB, 21474836480 bytes
255 heads, 32 sectors/track, 5140 cylinders, total  41942400 sectors
Units = sectors of 1 * 512 = 512 bytes
Disk identifier: 0x000da626

 Device    Boot      Start       End      Blocks   Id  System
 /dev/sda1   *         256     41942399    2097120   83  Linux root@petertest /root %
```

注：パーティションがシリンダ境界で終了（または開始）していない場合、fdiskユーティリティで警告が表示されることがあります。シリンダの概念は理論上のものであるため、これらの警告は無視して構いません。

推奨事項

システムとパーティション・レイアウトにディスクを追加する必要がある場合、パーティション・レイアウトを手動で作成してから、他の管理手順を使ってファイル・システムを追加したりファイル・システムを論理ボリューム・マネージャ（LVM）環境に組み込みます。これにより、管理ツールでデフォルト（ミスアライメント）のパーティションを使ってパーティションが作成されたり、最適なパーティション開始位置と終了位置をセクター番号で入力したときに拒否されるのを防止できます。

新しいLinux環境のインストール時に、システム・ディスクを別のシステムに一時的に追加するか、目的のパーティション・レイアウトを作成するのに必要なユーティリティが含まれたブートCDを使用するかのいずれかの方法で、システム・ディスクをパーティション化してみます。

ディスク・パーティションの使用を選択するときに、前述したLUNパーティション・アライメントの注意事項を考慮してください。

Sun ZFS Storage ApplianceのLUNを使用する場合は、1つのLUNに複数の物理パスを使用します。FC接続の場合は、2つの独立したSANファブリックのトポロジを使用することを推奨します。

LinuxにおけるSun ZFS Storage ApplianceのLUN構成のヒント

Linuxオペレーティング・システムでFC HBAポートのWWNを検出するのに特に定まった方法はありません。

/proc/scsi/adapter_type/nを使って/proc構造を検索する方法があります。ここで、adapter_typeはホスト・アダプタのタイプで、nはホスト・アダプタの番号です。

/sysディレクトリ構造でディレクトリfc_hostを検索するという方法もあります。

例：

```
[root@x4450-1 fc_host]# pwd
/sys/class/fc_host
[root@x4450-1 fc_host]# ls
host10 host9
[root@x4450-1 fc_host]# cat host9/port_name
0x2100001b320b5eb6
[root@x4450-1 fc_host]#
```

複数のパス接続を使用している場合、次の例に示すように、Linux multipathユーティリティを使用して、LUNに検出されたパスを表示します。

```
[root@x4450-1 fc_host]# multipath –ll
mpath1 (3600144f0f05e906c00004fe617540002) dm-6 SUN,Sun Storage 7210
size=128G features='0' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=enabled
| `- 9:0:0:1 sdf 8:80 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
`- 10:0:0:1 sdi 8:128 active ready running
mpath0 (3600144f0f05e906c00004fe6173d0001) dm-5 SUN,Sun Storage 7210
size=64G features='0' hwhandler='0' wp=rw
```

```
|-+- policy='round-robin 0' prio=1 status=enabled
| `- 9:0:0:0 sde 8:64 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
`- 10:0:0:0 sdh 8:112 active ready running
[root@x4450-1 fc_host]#
```

上記に示されているLUNが/dev/dm-5および/dev/dm-6として使用可能です。

Oracle VMを使用したディスク管理

Oracle VMは、OracleのワークロードとOracle以外のワークロードの両方でハードウェアとソフトウェアの完全なスタックを仮想化および管理する、統合型の仮想化ソリューション・ポートフォリオを提供します。Oracle VMには、Oracle VM Server for x86とOracle VM Managerが含まれています。

Oracle VM Managerを使用すると、Oracle VM Serverと仮想マシンを一元管理できます。Oracle VM Managerは、64ビットのOracle Linux 5.5以降で動作します。一般的なインストールは、5.5のLinuxカーネルと、Oracle VM Serverの管理と監視を実行するOracleソフトウェアが含まれた、1つのディスクで構成されます。Oracle Linuxのインストール・プロセスでは、一般的なLinuxインストール手順に従います。インストール・プロセスでインストール・ディスクを選択してパーティション化できるため、ディスクのパーティションは、SSDなどの512Bよりも大きいディスク・ブロックを使用するディスクとはアライメントされません。

ただし、これは一般的なインストール環境ではありません。パーティションのアライメントが重要な場合は、パーティションがすでに適切に配置されているディスクを使用し、Linuxのインストール・プロセスで、カーネルのインストールに使用するパーティションを選択します。

Oracleソフトウェアは、/u01ディレクトリ構造にインストールされますが、別のディスク・パーティションに配置するように選択することもできます。

Oracle VM Serverでは、Linuxタイプのディスク構造を使用してハイパーバイザーとカーネル環境が保存されます。ストレージ・リポジトリをローカル・ディスクから使用したり、NFSまたはiSCSIおよびFCをブロック・デバイスに使用したりできます。Oracle VMのストレージ・リポジトリにローカル・ディスクを使用するのは推奨されていません。この特定のサーバー・ハードウェアでの仮想マシンの使用が制限されるためです。

Oracle VM環境にクライアントOSをインストールするときには、前のセクションで説明している個々のOSに関する推奨事項に従ってください。

Oracle Automatic Storage Managementを使用したディスク管理

Oracle Automatic Storage Managementは、物理ディスクとファイル・システムの上に抽象化レイヤーを追加することで、これらのコンポーネントの管理を簡素化します。Oracle Automatic Storage Managementを使用して、管理者はディスク・グループを管理できます。ディスク・グループを使用すると、物理ディスクの障害から保護するとともに、手動でチューニングせずにI/Oのロードバランシングを実行できます。

ディスクは、Oracle Automatic Storage Managementによって全体として使用することも、OSのディスク・ラベル/パーティショニング・メカニズムを使用して部分的に使用することもできます。

Oracle Automatic Storage Managementの構成情報を取得するには、Oracle Enterprise Managerを使用するか、コマンド・ユーティリティasmcmdを使用します。この情報にアクセスするのに正しい環境変数を設定するには、次の例に示すようにoraenvコマンドを実行します。

```
 [oracle@x4450-1 ~]$ . oraenv
ORACLE_SID = [oracle] ? +ASM
The Oracle base for ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is /u01/app/oracle
[oracle@x4450-1 ~]$ asmcmd
ASMCMD> lsdsk -k
Total_MB  Free_MB   OS_MB  Name          Failgroup     Library  Label  UDID  Product  Redund   Path
 65536    65484   65536  UCMDATA_0000  UCMDATA_0000  System                        UNKNOWN  /dev/dm-5
 131072   131019  131072  UCMFRA_0000   UCMFRA_0000   System                        UNKNOWN  /dev/dm-6
ASMCMD> quit
[oracle@x4450-1 ~]$
```

asmcmdのサブコマンドlsdskで、ディスク・グループと、それらのディスク・グループを構成する物理ディスクのリストが表示されます。

このリストから、どのディスクまたはパーティションがOracle Automatic Storage Managementで使用されているのかを把握できます。パーティションが使用されている場合、ホストOSのfdiskコマンドを使用して、パーティションの開始位置が基盤となるLUNのブロック構造と適切にアライメントされているかどうかを確認します。

Oracle Solarisの場合、fdisk -W - <rawdisk device name>を使用します。

Linuxの場合、fdisk -lu <rawdisk device name>を使用します。

適切にアライメントされていないディスク・パーティションがASMディスク・グループに含まれている場合、ディスク・グループからこれらのパーティションを1つずつ削除するプロセスを実行してパーティション構造を修正し、パーティションをディスク・グループに追加し直すことができます。パーティションを削除して修正するために、余分なディスクまたはディスク・パーティションをディスク・グループに一時的に追加することが必要になることがあります。

Oracle Automatic Storage Managementを使用したアライメントに関する考慮事項

Oracle Automatic Storage Managementを使用する場合、このソフトウェアに対してディスク全体の使用を許可するのか、または管理者がディスク・パーティションを管理するのかという、パーティションのアライメントに影響する判断を1つだけ行う必要があります。

Oracle Automatic Storage Managementに対してディスク全体の使用を許可するというデフォルト・オプションのメリットは、管理者がパーティション・アライメントを心配せずに済むことです。Oracle Automatic Storage Managementでは最初のブロックのディスク/LUNが使用されるため、Oracle Automatic Storage Managementのすべてのi/Oがディスク/LUNブロック構造とアライメントされます。

デメリットは、ディスク/LUNがOracle Automatic Storage Managementによって使用されていることがfdiskでは認識されないことです。そのため、管理者がディスクは使用されていないとみなして、ディスクにパーティション構造を書き込んだために、Oracle Automatic Storage Managementによって作成されたデータ・ボリュームが破損するというリスクがあります。

ディスクのパーティションがOracle Automatic Storage Managementによって使用されることを指定するオプションを選択すると、ディスクが使用されていることが明確になります。デメリットは、各パーティションの開始位置が基盤となるディスク/LUNのブロック構造とアライメンされていることを確認する必要があることです。

いずれのオプションが最適なのかは、実際には、組織で行っている構成管理やIT部門のシステム管理者の人数によって異なります。構成情報と管理プロセス・ポリシーを適切に文書化するとともに、ITスタッフが規律を遵守することで、データ破損につながる操作が実行されるのを防止できます。

Microsoft Windowsオペレーティング・システムを使用したディスク管理

各種のMicrosoft Windowsオペレーティング・システムでディスク/LUNを管理する場合、次の情報を考慮してください。

Microsoft Windows Server 2008およびWindows Vista

Microsoft Windows Server 2008オペレーティング・システムとWindows Vistaオペレーティング・システムでは、パーティションのアライメントはデフォルトで1MB（4GBを超えるディスクの場合）に設定されます。次の参照先でレジストリに使用されている値を確認できます。

HKLM¥SYSTEM¥CurrentControlSet¥Services¥VDS¥Alignment

Microsoft Windows Server 2003以前

Microsoft Windows 2003 Server以前のWindows環境で作成されたパーティションは、定義上はアライメントされません。パーティション・テーブルのデフォルトの開始位置は32256バイトです。Windowsのダイナミック・ディスクの場合、パーティションのオフセットを確認するには、`dmdiag -v`コマンドを使用します。ベーシック・ディスクの場合、`wmic`コマンドを使用します。

作成したFC LUNにパーティションを作成するには、diskpartツールを使用します。次のテンプレートを使用すると、パーティションのオフセットを設定し、ドライブ文字を割り当てて、ファイル・アロケーション・ユニット・サイズ（`unit=<XXK>`）を指定できます。

```
Diskpart list disk select disk <DiskNumber> create partition primary align=<Offset_in_KB>
assign letter=<DriveLetter> format fs=ntfs unit=64K label="<label>" nowait
```

オフセット値には、128Kまたは128Kの倍数を使用します。選択するファイル・アロケーション・ユニット・サイズは、Windowsファイル・システムの使用方法に関するアプリケーションの推奨事項によって異なります。同じ値のZFSブロック・サイズを使用することを推奨します。

マイクロソフトでは、Windows 2003に提供されているディスク・パーティション・ツールのバージョンを更新しています（マイクロソフトのサポート技術情報923076を参照）。新しいバージョンのdiskpartを使用して、アライメントするブロック・サイズ、および必要なパーティションのサイズを指定できます。マイクロソフトのサポート技術情報929491'Windows Server 2003、Windows XP、およびWindows 2000で複数のディスクを使用するとディスク・パフォーマンスが予想より低下することがある'も参照してください。

VMwareを使用したディスク管理

WMware環境では、仮想マシンはVirtual Machine File System（VMFS）を使用し、仮想マシンのストレージ・ブロックはPCのfdiskパーティション内に作成されます。作成されたパーティションが、基盤となるSun ZFS Storage ApplianceのZFS構造とアライメントされていることが重要です。

ゲストOSのデータ・ファイル・システムは、VMFSファイル・システムの上にVirtual Machine DisK（VMDK）形式で格納され、このVMFSファイル・システムとアライメントする必要があります。

また、ゲスト・オペレーティング・システムでRaw Device Mappings（RDM）を使用できます。RDMオプションを使用したパーティション・アライメントは、前述のLUNをそのまま使用するOSの場合と同じです。

VMware ESX 3.0を使用する場合、fdiskツールを使用してESXのパーティションを管理する必要があります。VMware ESX4.0以降では、VMware vSphereクライアントを使用してVMFSパーティションを管理します。VMFSパーティションを作成する際のデフォルトは、VMware ESX 4.0バージョンとVMware ESX 5.0バージョンで異なります。次に、2つのVMFS形式を示します。

- VMFS3－MBRタイプのパーティション形式を引き続き使用しています。VMware ESX 4.0では、MBRのデフォルト・サイズは64Kです。VMware ESX 5.0では、MBR領域のサイズは1MBです。
- VMFS5－GPTタイプのパーティション形式を使用し、GPTブロックの開始位置のサイズは1MBです。

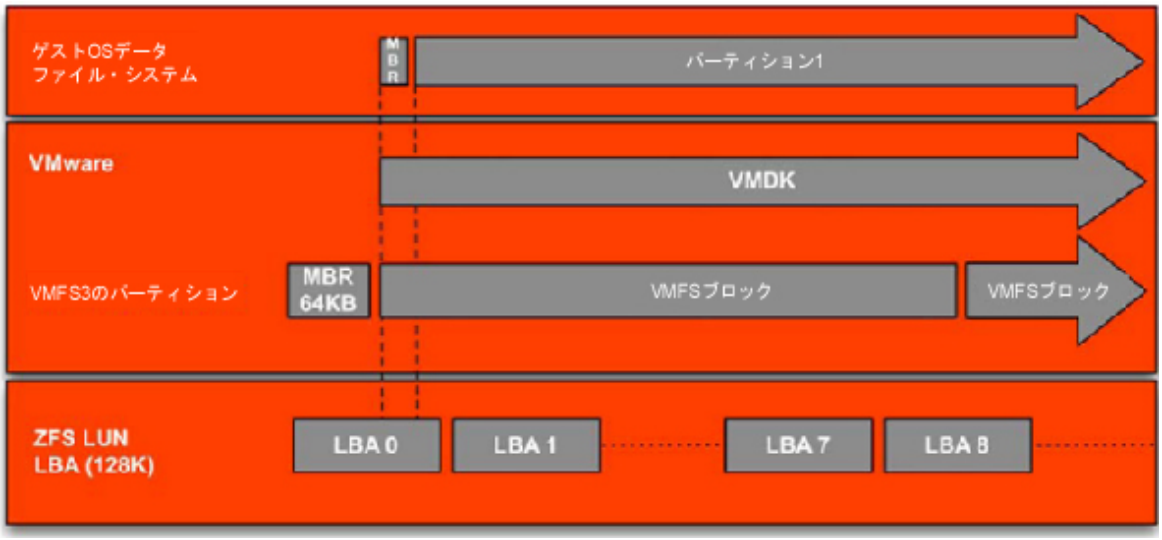

図10. VMwareデフォルトのVMFS3パーティション・レイアウト

常に1MBのサイズのMBR領域を使用することを推奨します。これにより、予約済みの領域が1MBという現在の業界標準にシステムを準拠させて、Sun ZFS Storage ApplianceのLUNのブロック構造にVMFSを常にアライメントさせることができます。

ゲストOSディスクのアライメントについては、前のセクションで説明している各オペレーティング・システムの推奨事項に従ってください。

VMwareには、OSの物理インスタンスを仮想インスタンスに変換したり、仮想環境間でインスタンスを移動するためのユーティリティが用意されています。このツールを使用して、ゲストOSのパーティション・レイアウトを最適なパーティション・レイアウトに修正できます。

VMware VMFSのブロック・サイズ

VMFSのブロック・サイズによってVMFSインスタンスの最大サイズが決まります。ゲストOS（ファイル・システム）で使用されるI/OサイズとVMFSのブロック・サイズの間には関係はありません。ゲストOSでの読取り操作と書込み操作は、VMkernelによって外部ストレージ・サブシステムに透過的に渡されます。そのため、使用するZFSブロック・サイズの値は、ゲストOSのファイル・システムとアプリケーションで使用されるI/Oサイズで判断する必要があります。

VMwareとNFS

VMwareでは、LUNをストレージとして使用する代わりに、格納するNFSファイル・システムをVMのVMDKファイルとしてマウントできます。これにより、VMFSレイヤーがバイパスされます。VMwareでは、Sun ZFS Storage Applianceの構成済みのNFSボリュームがマウントされます。VMware ESXホストを選択し、「Configurations」タブ→「Advanced Settings」→「NFS」を選択すると、VMware vSphereクライアントからVMwareのマウント・オプションを表示できます。

VMDKファイルの開始位置は、NFSファイル・システムが提供されるSun ZFS Storage Applianceのボリュームのブロックと常にアライメントされます。ゲストOSのパーティションの開始位置をSun ZFS Storage Applianceのブロック構造とアライメントすることが重要です。最初のパーティションに1MBのパーティション・オフセットを使用し、この1MBのブロックの開始位置にアライメントされている他のパーティションの開始セクターを使用すると、ゲストOSのI/OをSun ZFS Storage Applianceに確実にアライメントできます。

Sun ZFS Storage Applianceに共有を設定する場合、ZFSボリュームのブロック・サイズを使用する（仮想）アプリケーション環境に最適な値に設定していることを確認します。Sun ZFS Storage ApplianceのBUIを使って、共有のPropertiesウィンドウでプロパティ・オプション'Database record size'を使用します。

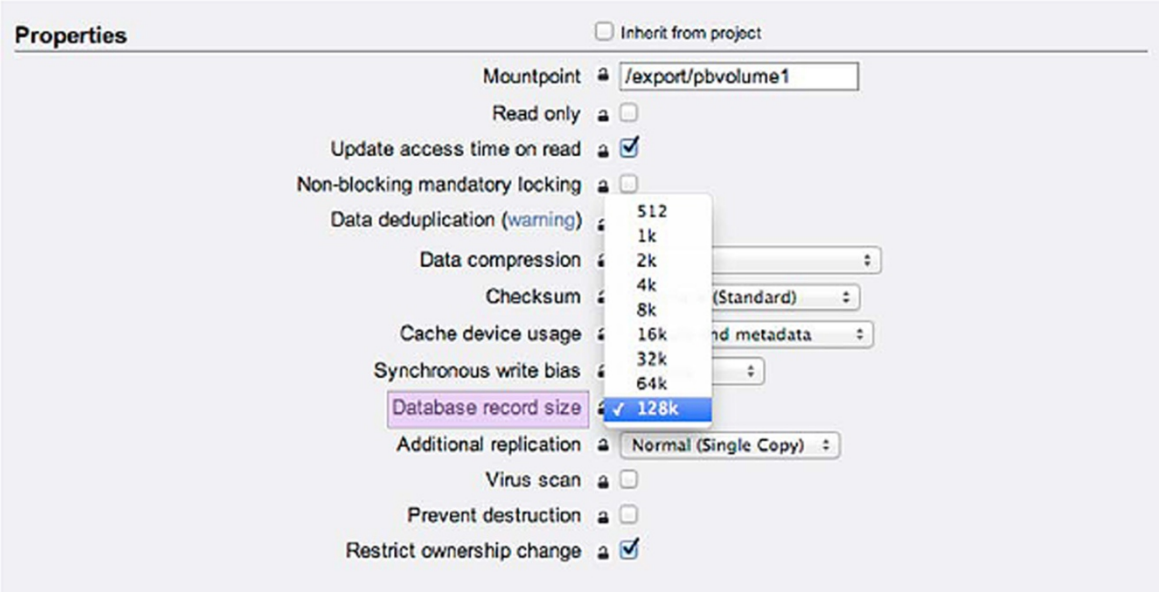

図11. Sun ZFS Storage ApplianceのBUIを使った共有レコード・サイズの設定

Sun ZFS Storage Applianceの分析ツールを使用したミスアライメントの診断

Sun ZFS Storage ApplianceのDTrace Analyticsオプションは、Sun ZFS Storage Applianceに接続されている各種ホストで生成されたI/Oワークロードを分析するための強力なツールです。ホストや操作のタイプなどでビューをドリルダウンするオプションを使って、ネットワーク、プロトコル、キャッシュ、CPU、およびディスク・サブシステムのワークロードを分析できます。

パーティションのミスアライメントのために、ストレージ・サブシステムに追加のI/O操作が発生するため、DTrace Analyticsを使用して、プロトコル・レベル（FCまたはiSCSI）で発行されたI/Oの数とディスク・サブシステムに対して発行されたI/Oの数を比較し、ミスアライメントの証拠を見つけることができます。ディスクのI/Oロードを生成しているアプリケーションによってホストが使用されている場合は、この比較は実行できません。ベンチマーク評価と同様に、ストレージ・サブシステムのワークロードの状態を分析するには、ホストのワークロードを把握しており、制御できる必要があります。キャッシュ・サブシステムで処理されて、測定時にディスクに到達しないI/Oがあれば、ホストのI/Oキャッシュ・システムで処理されるI/Oもあります。使用するRAID保護のタイプによって、論理書込みが物理的なディスクI/Oにどのように適用されるかが変わるため、書込みI/Oは使用できません。パリティ・ブロックやミラー・ブロックを書き込むには、追加のI/O操作が必要です。

ミスアライメントの問題を調べるのに最適な方法は、キャッシュ・サブシステムでは十分に生成されない、ランダムな読取りロードを生成できるI/Oベンチマーク・ツールを使用することです。キャッシュ・サブシステム内のメモリよりも大きいサイズのLUNを使用する必要があります。ランダムな読取りワークロードを使用し、読取り対象のブロックにワークロードがすでに書き込まれていることを確認します。ブロックに何も書き込まれていない場合は、読取りの完了がZFSファイル・システムから直接返されます。可能な場合は、ディスクのRAWデバイスを使用します。DTrace Analyticsをセットアップして、使用されているI/Oインタフェース（プロトコル）からの読取りとディスクへの読取りを監視します。

DTrace Analytics出力の次のスクリーンショットは、Oracle Solarisホストのファイバ・チャネルLUNを示したものです。最初の表示は、セクター128で始まるEFIラベル付きパーティションを示しており、2番目の表示はセクター34で始まるパーティションを示しています。グラフに示すように、最初の測定期間では、IOPが1850となっており、FC LUNのI/O数とディスクのI/O数に1対1の関係がみられます。

図12. それぞれセクター256と34で始まる、EFIラベル付きLUNのパーティションのI/O表示

2回目の測定期間では、FC LUNの各読取りで2つのディスク読取りI/Oが生成されています。そのため、1秒あたりのIOP数は、FCのI/OブロックをZFSのブロック・サイズとアライメントした場合よりも少なくなっています。

また、この例では、1850 IOPと1182 IOPの結果を比較し、ミスアライメントされたパーティションの読取りパフォーマンスが低下することも示されています。

結論

基盤となるディスクI/OサブシステムのNBSブロックとのパーティション・アライメントは、ホスト・アプリケーションで使用されるストレージ・サブシステムのパフォーマンス機能を最適に利用するために重要です。ディスク・ラベルとパーティション作成用のツールで使用されているデフォルトでは、最適化できないことが多いです。パーティションとファイル・システムの最適なレイアウトを作成するには、ブロックのI/Oスタック、パーティションの配置、およびパーティションと基盤となるストレージ・デバイスのナチュラル・ブロック・サイズとのアライメント関係を理解することが重要です。

使用するオペレーティング・システムでストレージ・デバイスのラベル付けとパーティション化に用意されているツールの特性を理解することも、同じように重要です。たとえば、CHSアドレッシング・メカニズムとLBAアドレッシング・メカニズムを比較して、使用の柔軟性が挙げられます。CHSアドレッシングのみサポートしているツールでは、適切にアライメントされたパーティションを作成するのは難しいです。

このことを理解して対処すると、ストレージ・デバイスのI/Oパフォーマンスをすぐに30〜50%向上できます。ボリュームあたりで使用するディスク・デバイスを増やす、またはアプリケーション・ボリュームを再構成してストレージ・デバイスに分散させるなどの、ハードウェアの追加の調査が伴う他の対処方法と比べて、パフォーマンスを大幅に向上できます。

計画が重要です。たとえば、LUNを作成する前にLUNの適切なナチュラル・ブロック・サイズ設定をあらかじめ決定しておくと、最適なアプリケーション・ボリュームやOSファイル・システムのレイアウトを作成するために、その都度変更を加えずに済みます。変更が発生した場合、現在のボリューム、ファイル・システム、およびパーティションを破棄する可能性が高いです。

付録：参考資料

- 『Sun ZFS Storage 7000システム管理ガイド』

  Part No. 820-4167-1x

  http://docs.oracle.com/cd/E40258_01/index.html

- 『Sun ZFS Storage Appliance Administration Guide』もSun ZFS Storage Applianceのヘルプ・コンテキストで参照できます。Sun ZFS Storage Applianceのヘルプ機能にアクセスするには、ブラウザ・ユーザー・インタフェースを使用します。

- Sun ZFS Storage Applianceのホワイト・ペーパーおよびテーマ別のリソース

  http://www.oracle.com/technetwork/jp/server-storage/sun-unified-storage/documentation/index.html

- Sun ZFS Storage Appliance製品情報

  http://www.oracle.com/jp/products/servers-storage/storage/nas/overview/index.html

- 『Oracle Solaris System Administration Guide』

  http://download.oracle.com/docs/cd/E19963-01/index.html

- Unified Extensible File Interface（UEFI）の情報

  http://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface

- GUIDパーティション・テーブルの情報

  http://en.wikipedia.org/wiki/GUID_Partition_Table

- マスター・ブート・レコードの情報

  http://en.wikipedia.org/wiki/Master_boot_record

- パブリック・ドメインのディスク・ユーティリティ例（パーティション管理のセクションを参照）
  http://ubcd.sourceforge.net/index.html

- マイクロソフトのサポート技術情報

  http://support.microsoft.com

ストレージのパフォーマンスを最大化する
パーティション・アライメント
2012年11月、バージョン1.0

著者：Peter Brouwer
Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

お問い合わせ窓口

Oracle Direct

TEL 0120-155-096
URL oracle.com/jp/direct

Oracle is committed to developing practices and products that help protect the environment

Copyright © 2012, Oracle and/or its affiliates.All rights reserved.
本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクル社は本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

OracleおよびJavaはOracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

IntelおよびIntel Xeonは Intel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。UNIXはX/Open Company, Ltd.によってライセンス提供された登録商標です。

0611

Hardware and Software, Engineered to Work Together